OpenVPN COMO

Introducción

Este documento describe cómo configurar OpenVPN en un entorno típico Casa-Oficina. Aunque este COMO presenta en profundidad la configuración con ejemplos, se pueden encontrar unos más sencillos en la sección de ejemplos de la página del manual.

Este COMO también está disponible en otros formatos:

- PDF (inglés)
- PostScript (inglés)

Otros artículos y documentación

Hay muchos otros artículos y COMOs sobre cómo configurar OpenVPN en distintos entornos.

Tipos básicos de túneles

Hay dos tipos básicos de túneles que se pueden crear con OpenVPN:

- **Túnel IP** -- usado para encaminar tráfico IP punto-a-punto sin broadcast. Bastante más eficiente que un puente ethernet y más faciles de configurar. Este COMO cubre esta clase de túneles.
- **Puente ethernet** -- se pueden usar para encapsular tanto protocolos IP como no-IP. Este tipo de túnel es apropiado para aplicaciones que se comunican utilizando difusión (broadcast), tales como la red de Windows y juegos de área local (LAN). Son bastante más dificiles de configurar. Mini-COMO para puentes ethernet.

Túnel IP COMO

Dado el elevado número de asuntos relacionados para configurar firewalls, VPNs y NAT se tratará de describir la configuración de un sistema completo en lugar de describir únicamente la configuración de la VPN.

En nuestro ejemplo, tanto las redes privadas de Casa como la del Trabajo se unen a Internet por medio de dos puertas de enlace, cada una de las cuales tienen una dirección IP pública. Cada máquina que actua como puerta de enlace tiene dos interfaces de red, una conectada a la red privada y la otra conectada a Internet. Las puertas de enlace dan soporte a los servicios NAT, firewall y VPN para las máquinas de las redes privadas. Tanto la configuración de Casa como la de la Oficina son casi simétricas exceptuando que la Oficina tiene una dirección IP fija mientras que la de Casa tiene una dirección IP dinámica (DHCP).

Los ficheros de configuración de los siguientes ejemplos estan también disponibles en el paquete de OpenVPN.

Parámetros de configuración para la red de Casa y la Oficina

	Casa	**Oficina**
Subred ethernet local (Dirección privada)	10.0.1.0/24	10.0.0.0/24
Extremo del túnel (Dirección privada)	10.1.0.2	10.1.0.1
Puerta de enlace OpenVPN (Dirección pública)	cliente DHCP, no necesita ser especificada	1.2.3.4

Instalando OpenVPN

Si su sistema no dispone de la biblioteca OpenSSL debe descargarla e instalarla.

Si quiere utilizar la compresión sobre el enlace VPN, o desea instalar OpenVPN como un paquete RPM instale la biblioteca LZO.

Si utiliza un Linux 2.2 o anteror descargue el controlador TUN/TAP. Los usuarios de Linux 2.4.7 o superior deberían tener el controlador TUN/TAP ya incluido en su kernel. Usuarios

de Linux 2.4.0 -> 2.4.6 deberían leer la advertencia al final del fichero INSTALL.

Descargue ahora la última release de OpenVPN:

> http://prdownloads.sourceforge.net/openvpn/openvpn-1.6.0.tar.gz

Instalar desde paquete tar

Descomprima el paquete:

```
gzip -dc openvpn-1.6.0.tar.gz | tar xvf -
```

Compilar OpenVPN:

```
cd openvpn-1.6.0
./configure
make
make install
```

Si no se descargó la biblioteca LZO, añada **--disable-lzo** al comando **configure**. Se pueden habilitar otras opciones como el soporte para **pthread** (**./configure --enable-pthread**) para mejorar la latencia durante los intercambios dinámicos de clave SSL/TLS. El comando

```
./configure --help
```

muestra todas las opciones de configuración.

Instalación a partir de paquete RPM

Primero construya el fichero RPM. Ésto requiere que las bibliotecas de OpenSSL, pthread y LZO esten instaladas. Normalmente solo la bibliotecta LZO necesita ser descargada e instalada explícitamente; las otras bibliotecas estan presentes en el sistema por defecto en la mayoría de las distribuciones de Linux.

```
rpmbuild -tb openvpn-1.6.0.tar.gz
```

El proceso de construción del RPM generará mucha traza de salida. Si la construcción tiene éxito habrá una nota cerca del fichal de la salida, indicando el nombre del fichero RPM binario generado. Instale el paquete RPM binario con el comando:

```
rpm -Uvh binary-RPM-file
```

Configuración del controlador TUN/TAP

Pasos de configuración a realizar una única vez

Si esta usando Linux 2.4.7 o superior, es probable que el controlador TUN/TAP este ya incluido en el kernel. Puede confirmarlo con el comando

```
locate if_tun.h
```

esto debe mostrar un fichero como **/usr/include/linux/if_tun.h**.

Para Linux 2.4.7 o superior, si instaló desde paquete tar, teclee el siguiente comando para crear el nodo del dispositivo TUN/TAP (puede saltarse este paso is ha instalado desde RPM, ya que el instalador de RPM lo realiza automáticamente):

```
mknod /dev/net/tun c 10 200
```

Si está usando Linux 2.2 debe descargar la versión 1.1 del modulo del kernel TUN/TAP y seguir las instrucciones de instalación.

Pasos de configuración a realizar cada vez que se arranque

En Linux antes de usar OpenVPN, o cualquier otro programa que utilice dispositivos TUN/TAP, debe cargar el modulo del kernel TUN/TAP:

```
modprobe tun
```

y habilitar IP forwarding:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Configure el Firewall y NAT

Esta sección supone que esta usando Linux 2.4 con un firewall **iptables**. Se presenta un ejemplo de configuración para el firewall que realiza NAT para las maquinas de la red privada para permitir el acceso a Internet, estratificar las conexiones de salida y soporte OpenVPN:

sample-config-files/firewall.sh

```
#!/bin/bash

# Un ejemplo de firewall que tenga en cuenta OpenVPN.

# eth0 está conectado a Internet.
# eth1 está conectado a la subred privada.

# Cambie esta subred para que se corresponda con su subred
# ethernet privada.  Casa usará 10.0.1.0/24 y la
# Oficina usará 10.0.0.0/24.
PRIVATE=10.0.0.0/24

# Direccion de loopback
LOOP=127.0.0.1

# Borrar reglas iptables antiguas
# y temporalmente bloquear el tráfico.
iptables -P OUTPUT DROP
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -F

# Establecer las políticas por defecto
iptables -P OUTPUT ACCEPT
iptables -P INPUT DROP
iptables -P FORWARD DROP

# Evitar que los paquetes externos usen la dirección de loopback
iptables -A INPUT -i eth0 -s $LOOP -j DROP
iptables -A FORWARD -i eth0 -s $LOOP -j DROP
iptables -A INPUT -i eth0 -d $LOOP -j DROP
iptables -A FORWARD -i eth0 -d $LOOP -j DROP

# Cualquer cosa que venga de Internet debería tener una dirección de Internet real
iptables -A FORWARD -i eth0 -s 192.168.0.0/16 -j DROP
iptables -A FORWARD -i eth0 -s 172.16.0.0/12 -j DROP
iptables -A FORWARD -i eth0 -s 10.0.0.0/8 -j DROP
iptables -A INPUT -i eth0 -s 192.168.0.0/16 -j DROP
iptables -A INPUT -i eth0 -s 172.16.0.0/12 -j DROP
iptables -A INPUT -i eth0 -s 10.0.0.0/8 -j DROP

# Bloquear paquetes NetBios salientes (si tiene máquinas windows en
# la subred privada).  Ésto no afecta al tráfico NetBios
# que circula por el túnel VPN, pero detendrá a las maquinas
# windows locales de mandar mensajes de broadcast
# a Internet.
iptables -A FORWARD -p tcp --sport 137:139 -o eth0 -j DROP
iptables -A FORWARD -p udp --sport 137:139 -o eth0 -j DROP
iptables -A OUTPUT -p tcp --sport 137:139 -o eth0 -j DROP
iptables -A OUTPUT -p udp --sport 137:139 -o eth0 -j DROP

# Comprobar la validez de la dirección origen de los paquetes salientes a Internet
iptables -A FORWARD -s ! $PRIVATE -i eth1 -j DROP

# Permitir loopback local
iptables -A INPUT -s $LOOP -j ACCEPT
iptables -A INPUT -d $LOOP -j ACCEPT

# Permitir pings entrantes (pueden desabilitarse)
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

# Permitir servicios tales como www y ssh (pueden desabilitarse)
iptables -A INPUT -p tcp --dport http -j ACCEPT
iptables -A INPUT -p tcp --dport ssh -j ACCEPT

# Permitir paquetes entrantes a OpenVPN
# Duplicar la línea inferior por cada
# túnel OpenVPN, cambiando --dport n
# para que encaje con el puerto UDP de OpenVPN.
#
# En OpenVPN, el número de puerto se
# control con la opción --port n.
# Si pone esta opción en el fichero de configuración,
# puede eliminar los caracteres iniciales '--'
#
# Si está usando el firewall con estado
# (consulte el OpenVPN COMO),
# entonces comente la línea de abajo.

iptables -A INPUT -p udp --dport 1194 -j ACCEPT

# Permitir paquete del dispositivo TUN/TAP.
# Cuando OpenVPN está ejecutando en modo seguro,
# autenticará los paquetes previos a
# su llegada en el interfaz
# tun o tap.  Por lo tanto, no es
# necesario añadir ningun filtro aqui,
# a menos que quiera restringir el
```

```
# tipo de paquete que puedan circular por
# el túnel.

iptables -A INPUT -i tun+ -j ACCEPT
iptables -A FORWARD -i tun+ -j ACCEPT
iptables -A INPUT -i tap+ -j ACCEPT
iptables -A FORWARD -i tap+ -j ACCEPT

# Permitir paquetes de subredes privadas
iptables -A INPUT -i eth1 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT

# Mantener el estado de las conexiones locales y las subredes privadas
iptables -A OUTPUT -m state --state NEW -o eth0 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state NEW -o eth0 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Enmascarar la subred local
iptables -t nat -A POSTROUTING -s $PRIVATE -o eth0 -j MASQUERADE
```

OpenVPN ofrece opciones adicionales a la configuración del firewall:

- Si ambos extremos OpenVPN referencian al contrario con la opción explicita **--remote**, y los firewalls dependiendes del estado tienen soporte para gestionar conexiones UDP (tales como **iptables**) existentes entre ambos extremos, es posible ejecutar OpenVPN sin ninguna regla explícita de firewall o si ambos extremos crean pings regularmente al otro extremo para mantener la conexion activa. Para hacer esto, simplemente ejecute OpenVPN con la opción **--remote *peer***, y especifique **--ping 15** para asegurar el flujo de paquetes en el túnel al menos una vez cada 15 segundos.
- La opción anterior es menos conveniente si uno de los extremos cambia su dirección IP frecuentemente, tales como DHCP o un extremo que utiliza un enlace telefónico. Por estos motivos, la configuración del firewall del ejemplo anterior permite a los paquetes entrantes al puerto 1194 UDP (puerto por defecto de OpenVPN) desde cualquier dirección IP. Esto debe considerarse seguro en cualquier de los modos seguros de OpenVPN, ya que todos los paquetes procedentes del túnel deben pasar un test de autenticación o serán descartados.
- Si elige el abrir completamente el puerto UDP de OpenVPN, como en el ejemplo de configuración del firewall anterior, lo mismo le interesa utilizar la opción **--tls-auth** para realizar una doble autenticación en el canal de control TLS, usando tanto la clave RSA como la contraseña secreta pre-compartida como una segunda línea de defensa contra DoS o ataques activos. Para más información de la opción **--tls-auth**, consulte la página man de openvpn.

Construcción de los certificados y claves RSA

OpenVPN tienes dos modos considerados seguros, uno basado en SSL/TLS usando certificados y claves RSA, el otro basado en claves estáticas pre-compartidas. Mientras que SSL/TLS + claves RSA es indiscutiblemente la opción más segura, las claves estáticas tienen la ventaja de la simplicidad. Si desea usar claves RSA, continue leyendo. Para claves estáticas, salte hacia delante a la sección constuir claves estáticas pre-compartidas.

Se van a contruir certificados y claves RSA utilizando el comando **openssl**, incluido en la distribución de la biblioteca OpenSSL.

Los certificados RSA son claves públicas que también tienen otros campos seguros insertados en ellos tales como el **Nombre Común** o la **dirección email** del propietario del certificado. OpenVPN provee la posibilidad de escribir scripts para probar estos campos antes de la autenticación. Para más información, consulte la opción **--tls-verify** en la página del manual de openvpn.

En el ejemplo se seguirá la convención de **apache** de usar la extensión de fichero **.crt** para denotar ficheros de certificados y la extensión **.key** para denotar ficheros de clave privada. Las claves privadas deben mantenerse protegidas siempre. Los ficheros con los certificados pueden publicarse libremente o compartirse.

Elija una máquina como por ejemplo la Oficina para que sea la máquina gestora de claves.

Primero edite el fichero **/usr/share/ssl/openssl.cnf** (este fichero puede estar en un sitio diferente, así que utilice **locate openssl.cnf** para encontrarlo).

Quizá quiera realizar cambios en el fichero:

- Haga un directorio que sirva como espacio de trabajo para las claves y cambie **dir** para que apunte a él.
- Considere incrementar **default_days** para que la VPN no deje de funcionar misteriosamente exactamente después de un año.
- Establezca **certificate** y **private_key** para que apunte al certificado maestro de la Autoridad de Certificación y la clave privada que se va a generar ahora. En los ejemplos de abajo, se asume que el certificado de la Autoridad de Certificación se llama **my-ca.crt** y la clave privada de la Autoridad de Certificación se llama **my-ca.key**.

- Observe los ficheros **index.txt** y **serial**. Inicialice **index.txt** para que esté vacio y **serial** para conter un número de serie inicial, como por ejemplo el **01**.
- Si usted es un paranóico en el tamaño de las claves, incremente **default_bits** a 2048. OpenVPN no tendrá problemas en manejar una clave RSA de 2048 bits RSA si ha compilador OpenVPN con soporte **pthread**, para habilitar el procesamiento en segundo plano de claves RSA. Puede usar claves mayores incluso sin soporte **pthread**, pero observará cierta degradación de la latencia en el túnel durante la negociación de las claves SSL/TLS. Para consultar un buen artículo sobre qué tamaño de clave RSA escoger, consulte el número de abril de 2002 del diario Crypto-Gram de Bruce Schneier.

Después de que **openssl.cnf** haya sido modificado, cree su Autoridad de Certificación maestra, un par certificado/clave privada:

```
openssl req -nodes -new -x509 -keyout my-ca.key -out my-ca.crt -days 3650
```

Esto creará una Autoridad de Certificación maestra, un par certificado/clave privada, válida por 10 años.

Ahora cree un par certificado/clave privada para Casa y otro para la Oficina:

```
openssl req -nodes -new -keyout office.key -out office.csr
openssl ca -out office.crt -in office.csr
openssl req -nodes -new -keyout home.key -out home.csr
openssl ca -out home.crt -in home.csr
```

Ahora copie **home.crt**, **home.key** y **my-ca.crt** a Casa a través de un canal seguro, aunque actualmente solo los ficheros **.key** deben considerarse no-públicos.

Ahora establezca los parámetros Diffie Hellman en la Oficina con el siguiente comando:

```
openssl dhparam -out dh1024.pem 1024
```

Incremente el número de bits de 1024 a 2048 si lo incrementó también en **openssl.cnf**.

Para el paranóico, considere omitir la opción **-nodes** en los comandos **openssl** de arriba. Esto producira que cada clave privada se cifre con un password, haciendo las claves seguras incluso si alguien entra en el servidor y roba los ficheros con la clave privada. La parte negativa de esto es que cada vez que ejecute OpenVPN deberá teclear el password. Para más información consulte la opción **--askpass** en la página man de openvpn.

Si encuentra la gestión manual de claves RSA confusa, OpenVPN también soporta interoperar con cualquier herramienta de gestión de certificados X509 o servicio incluyendo las CAs comerciales tales como Thawte o Verisign. Pruebe el proyecto OpenCA como un ejemplo de qué se está haciendo para la gestión de certificados/claves en el mundo Open Source.

Además, la distribución de OpenVPN contiene un conjunto pequeño de scripts que pueden usarse para simplificar la gestión de claves y certificados RSA.

Nota importante para el uso de Autoridades de Certificación (CAs) comerciales con OpenVPN

Debe notar que el modo de seguridad de OpenVPN en modo SSL/TLS está orientado hacia usuarios que van a generar sus propios certificados raíz y, por lo tanto, su propia CA. En el modo SSL/TLS, OpenVPN autentica el extremo comprobando que el certificado ofrecido está firmado por la CA especificada en la opción **--ca**. Como en un servidor web con soporte SSL, la seguridad del modo SSL/TLS de OpenVPN reside en la dificultad de falsificar la firma del certificado raíz.

Este mecanismo de autenticación funciona perfectamente si ha generado su propio certificado raíz, pero presenta un problema si desea usar el certificado raíz de una CA comercial, tal como Thawte. Si, por ejemplo, especifica el certificado raíz de Thawte en la opción**--ca**, cualquier certificado firmado por Thawte podrá ahora autenticarse con el extremo OpenVPN que seguramente no es lo que desea.

Afortunadamente hay una solucion para este problema en la opción**--tls-verify**. Esta opción permitirá ejecutar un comando para comprobar el contenido del certificado, para afinar la eleccion de qué certificados se permiten y cuales no. Consulte el script **verify-cn** en el subdirectorio sample-scripts para ver el ejemplo de cómo hacer esto, y además consulte en la página del manual la opción **--tls-verify**.

Nota importante para un posible ataque "Man-in-the-Middle" si los clientes no verifican el certificado del servidor al que estan conectandose.

Para evitar un posible ataque "Man-in-the-Middle" donde un cliente autorizado intenta conectarse con otro cliente suplantando la identidad del servidor, asegurese de forzar algún tipo de verificación de certificado por parte de los clientes. Actualmente hay cuatro maneras diferentes de realizar esto, mostradas en orden de preferencia:

- Cree los certificados del servidor con el script **build-key-server** (consulte la documentación easy-rsa para más información). Esto marcará el certificado como un

certificado sólo-servidor estableciendo **nsCertType**=server. Ahora añada la siguiente línea a la configuración del cliente:

```
ns-cert-type server
```

Esto bloqueará a los clientes al conectarse a cualquier servidor que no tenga el atributo **nsCertType**=server en su certificado, incluso si el certificado ha sido firmado por el fichero de la **ca** que se ha indicado en la configuración de OpenVPN.

- Use la directiva **tls-remote** en el cliente para aceptar/rechazar la conexión del servidor basandose en el el Nombre Común del certificado del servidor.
- Use el script **tls-verify** o un plugin para aceptar/rechazar la conexión del servidor basandose en un test a medida del certificado X509 del servidor a partir del contenido del asunto.
- Firme los certificados de servidor con una CA y los certificados de los clientes con otra CA diferente. La configuración de la directiva **ca** del cliente debe referenciar el fichero de la CA firmante del servidor, mientras que en la configuración del servidor la directiva **ca** debe referenciar el fichero de la CA firmante del cliente.

Fichero de configuración usando el modo SSL/TLS y certificados/claves RSA

En el ejemplo, se va a usar un fichero de configuración de OpenVPN. OpenVPN permite pasar opciones en la línea de mandatos o en uno o más ficheros de configuración. Las opciones de los ficheros de configuración pueden omitir los caracteres iniciales "--" necesarios para las opciones de la línea de mandatos.

Modifique los siguientes ficheros de configuración:

---

sample-config-files/tls-office.conf

```
#
# Fichero de configuración ejemplo de OpenVPN para
# la Oficina usando el modo SSL/TLS y certificados/claves RSA.
#
# '#' o ';' pueden usarse para delimitar comentarios.

# Usar un dispositivo tun dinámico.
# Para Linux 2.2 o SSOO no-Linux,
# lo mismo desea utilizar un número
# explícitamente como por ejemplo "tun1".
# OpenVPN también soporta dispositivos ethernet
# virtuales "tap".
dev tun

# 10.1.0.1 es nuestro extremo local VPN (Oficina).
# 10.1.0.2 es nuestro extremo remoto VPN (Casa).
ifconfig 10.1.0.1 10.1.0.2

# Script que establecerá las rutas
# una cuando la VPN esté activa.
up ./office.up

# En el intercambio de claves SSL/TLS, la Oficina
# asumira el rol de servidor y Casa
# asumirá el rol de cliente.
tls-server

# Parámetros Diffie-Hellman (sólo para tls-server)
dh dh1024.pem

# Fichero de la Autoridad de Certificación (CA)
ca my-ca.crt

# Nuestro certificado/clave pública
cert office.crt

# Nuestra clave privada
key office.key

# OpenVPN utiliza el puerto 1194 UDP por defecto.
# Cada túnel OpenVPN debe usar
# un número de puerto diferente.
# lport o rport pueden usarse
# para denotar diferentes puertos
# para local y remoto.
; port 1194

# Rebajar UID y GID a
# "nobody" después de la inicialización
# para más seguridad.
; user nobody
```

```
; group nobody

# Si compila OpenVPN con
# compresión LZO, descomente
# la siguiente línea.
; comp-lzo

# Enviar un ping UDP al extremo remoto una vez
# cada 15 segundos para mantener
# el estado la conexión en el firewall
# activa.  Descomente esto
# si está usando un firewall con
# estado.
; ping 15

# Descomente ésto para una detección más fiable cuando el sistema
# pierde su conexión.  Por ejemplo, conexiones telefónicas o portátiles que
# se desplazan a otros sitios.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Nivel de información.
# 0 -- callado excepto en errores fatales.
# 1 -- casi callado, pero mostrar errores no-fatales de red.
# 3 -- información media, para funcionar normalmente.
# 9 -- mucha información, útil para resolución de problemas
verb 3
```

---

sample-config-files/office.up

```
#!/bin/bash
route add -net 10.0.1.0 netmask 255.255.255.0 gw $5
```

---

sample-config-files/tls-home.conf

```
#
# Fichero de configuración ejemplo de OpenVPN para
# Casa usando el modo SSL/TLS y certificados/claves RSA.
#
# '#' o ';' pueden usarse para delimitar comentarios.

# Usar un dispositivo tun dinámico.
# Para Linux 2.2 o SSOO no-Linux,
# lo mismo desea utilizar un número
# explícitamente como por ejemplo "tun1".
# OpenVPN también soporta dispositivos ethernet
# virtuales "tap".
dev tun

# Nuestro extremo OpenVPN es la puerta de enlace de la oficina.
remote 1.2.3.4

# 10.1.0.2 es nuestro extremo local VPN (Casa).
# 10.1.0.1 es nuestro extremo remoto VPN (Oficina).
ifconfig 10.1.0.2 10.1.0.1

# Script que establecerá las rutas
# una cuando la VPN esté activa.
up ./home.up

# En el intercambio de claves SSL/TLS, la Oficina
# asumira el rol de servidor y Casa
# asumirá el rol de cliente.
tls-client

# Fichero de la Autoridad de Certificación (CA)
ca my-ca.crt

# Nuestro certificado/clave pública
cert home.crt

# Nuestro certificado/clave pública
key home.key

# OpenVPN utiliza el puerto 1194 UDP por defecto.
# Cada túnel OpenVPN debe usar
# un número de puerto diferente.
# lport o rport pueden usarse
# para denotar diferentes puertos
```

```
# para local y remoto.
; port 1194

# Rebajar UID y GID a
# "nobody" después de la inicialización
# para más seguridad.
; user nobody
; group nobody

# Si compila OpenVPN con
# compresión LZO, descomente
# la siguiente línea.
; comp-lzo

# Enviar un ping UDP al extremo remoto una vez
# cada 15 segundos para mantener
# el estado la conexión en el firewall
# activa.  Descomente esto
# si está usando un firewall con
# estado.
; ping 15

# Descomente ésto para una detección mas fiable cuando el sistema
# pierde su conexión.  Por ejemplo, conexiones telefónicas o portátiles que
# se desplazan a otros sitios.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Nivel de información.
# 0 -- callado excepto en errores fatales.
# 1 -- casi callado, pero mostrar errores no-fatales de red.
# 3 -- información media, para funcionar normalmente.
# 9 -- mucha información, útil para resolución de problemas
verb 3
```

---

sample-config-files/home.up

```
#!/bin/bash
route add -net 10.0.0.0 netmask 255.255.255.0 gw $5
```

Construir una clave estática pre-compartida

En contraste a la gestion de claves RSA, el usar una clave estática pre-compartida tiene el beneficio de ser más simple. La parte negativa de usar claves estáticas es que se pierde la noción de *perfecta seguridad en el futuro (perfect forward secrecy)*, significando que, si un atacante roba la clave estática, cualquier cosa que haya sido cifrada con esa clave estará comprometida.

Genere una clave estática con el siguiente comando:

```
openvpn --genkey --secret static.key
```

La clave estática está formateada en ascii y tiene un aspecto como éste:

```
-----BEGIN OpenVPN Static key V1-----
e5e4d6af39289d53
171ecc237a8f996a
97743d146661405e
c724d5913c550a0c
30a48e52dfbeceb6
e2e7bd4a8357df78
4609fe35bbe99c32
bdf974952ade8fb9
71c204aaf4f256ba
eeda7aed4822ff98
fd66da2efa9bf8c5
e70996353e0f96a9
c94c9f9afb17637b
283da25cc99b37bf
6f7e15b38aedc3e8
e6adb40fca5c5463
-----END OpenVPN Static key V1-----
```

Un fichero de clave estática OpenVPN contiene suficiente entropía como para almacenar tanto una clave cifradora de 512 bits como una clave HMAC de 512 bits para autenticación.

Copie **static.key** al otro extremo por medio de un medio seguro tal como **scp** o copiar-pegar en una conexión **ssh**.

Fichero de configuración usando una clave estática pre-compartida

En el ejemplo, se va a usar un fichero de configuración de OpenVPN. OpenVPN permite pasar opciones en la línea de comandos o en uno o más ficheros de configuración. Las opciones de los ficheros de configuración pueden omitir los caracteres iniciales "--" necesarios para las opciones de la línea de comandos.

Modifique los siguientes ficheros de configuración:

---

sample-config-files/static-office.conf

```
#
# Fichero de configuración ejemplo de OpenVPN para
# la Oficina usando una clave estática pre-compartida.
#
# '#' o ';' pueden usarse para delimitar comentarios.

# Usar un dispositivo tun dinámico.
# Para Linux 2.2 o SSOO no-Linux,
# lo mismo desea utilizar un número
# explícitamente como por ejemplo "tun1".
# OpenVPN también soporta dispositivos ethernet
# virtuales "tap".
dev tun

# 10.1.0.1 es nuestro extremo local VPN (Oficina).
# 10.1.0.2 es nuestro extremo remoto VPN (Casa).
ifconfig 10.1.0.1 10.1.0.2

# Script que establecerá las rutas
# una cuando la VPN esté activa.
up ./office.up

# Nuestra clave estática pre-compartida
secret static.key

# OpenVPN utiliza el puerto 1194 UDP por defecto.
# Cada túnel OpenVPN debe usar
# un número de puerto diferente.
# lport o rport pueden usarse
# para denotar diferentes puertos
# para local y remoto.
; port 1194

# Rebajar UID y GID a
# "nobody" después de la inicialización
# para más seguridad.
; user nobody
; group nobody

# Si compila OpenVPN con
# compresión LZO, descomente
# la siguiente línea.
; comp-lzo

# Enviar un ping UDP al extremo remoto una vez
# cada 15 segundos para mantener
# el estado la conexión en el firewall
# activa.  Descomente esto
# si está usando un firewall con
# estado.
; ping 15

# Descomente ésto para una detección mas fiable cuando el sistema
# pierde su conexión.  Por ejemplo, conexiones telefónicas o portátiles que
# se desplazan a otros sitios.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Nivel de información.
# 0 -- callado excepto en errores fatales.
# 1 -- casi callado, pero mostrar errores no-fatales de red.
# 3 -- información media, para funcionar normalmente.
# 9 -- mucha información, útil para resolución de problemas
verb 3
```

---

sample-config-files/office.up

```
#!/bin/bash
route add -net 10.0.1.0 netmask 255.255.255.0 gw $5
```

---

sample-config-files/static-home.conf

```
#
# Fichero de configuración ejemplo de OpenVPN para
# Casa usando una clave estática pre-compartida.
#
# '#' o ';' pueden usarse para delimitar comentarios.


# Usar un dispositivo tun dinámico.
# Para Linux 2.2 o SSOO no-Linux,
# lo mismo desea utilizar un número
# explícitamente como por ejemplo "tun1".
# OpenVPN también soporta dispositivos ethernet
# virtuales "tap".
dev tun

# Nuestro extremo OpenVPN es la puerta de enlace de la oficina.
remote 1.2.3.4

# 10.1.0.2 es nuestro extremo local VPN (Casa).
# 10.1.0.1 es nuestro extremo remoto VPN (Oficina).
ifconfig 10.1.0.2 10.1.0.1

# Script que establecerá las rutas
# una cuando la VPN esté activa.
up ./home.up

# Nuestra clave estática pre-compartida
secret static.key

# OpenVPN utiliza el puerto 1194 UDP por defecto.
# Cada túnel OpenVPN debe usar
# un número de puerto diferente.
# lport o rport pueden usarse
# para denotar diferentes puertos
# para local y remoto.
; port 1194

# Rebajar UID y GID a
# "nobody" después de la inicialización
# para más seguridad.
; user nobody
; group nobody

# Si compila OpenVPN con
# compresión LZO, descomente
# la siguiente línea.
; comp-lzo

# Enviar un ping UDP al extremo remoto una vez
# cada 15 segundos para mantener
# el estado la conexión en el firewall
# activa.  Descomente esto
# si está usando un firewall con
# estado.
; ping 15

# Descomente ésto para una detección mas fiable cuando el sistema
# pierde su conexión.  Por ejemplo, conexiones telefónicas o portátiles que
# se desplazan a otros sitios.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Nivel de información.
# 0 -- callado excepto en errores fatales.
# 1 -- casi callado, pero mostrar errores no-fatales de red.
# 3 -- información media, para funcionar normalmente.
# 9 -- mucha información, útil para resolución de problemas
verb 3
```

---

sample-config-files/home.up

```
#!/bin/bash
route add -net 10.0.0.0 netmask 255.255.255.0 gw $5
```

Arrancar la VPN en modo SSL/TLS

En Casa, arranque la VPN con el comando:

```
openvpn --config tls-home.conf
```

En la Oficina, arranque la VPN con el comando:

```
openvpn --config tls-office.conf
```

Arrancar la VPN en modo clave estática

En Casa, arranque la VPN con el comando:

```
openvpn --config static-home.conf
```

En la Oficina, arranque la VPN con el comando:

```
openvpn --config static-office.conf
```

Pruebe la VPN

En Casa, compruebe la VPN realizando un ping a la Oficina a traves del túnel:

```
ping 10.1.0.1
```

En la Oficina, compruebe la VPN realizando un ping a Casa a traves del túnel:

```
ping 10.1.0.2
```

Si estas pruebas fallan, puede re-editar los ficheros de configuración y poner el nivel de información a 8, lo cual producirá información de depuración mucho más detallada. También consulte la FAQ para más información y resolución de problemas.

Si estas pruebas tienen éxito, ahora intente realizar un ping a traves del túnel usando máquinas en la red privada que no sean las puerdas de enlace, para probar el rutado de paquetes. Basicamente cualquier máquina en la subred **10.0.1.0/24** debería poder acceder a cualquier máquina en la subred **10.0.0.0/24** y viceversa.

Si eso funciona, ¡ enhorabuena ! Si no, lo mismo desea buscar los archivos de la lista de correo de OpenVPN para ver si alguien más tiene un problema similar. Si no encuentra una solución a su problema ahí considere escribir a la lista **openvpn-users**.

Proporcionar soporte para DHCP en la VPN

Si recuerda, en nuestro ejemplo de configuración de red, Casa tiene una dirección IP que puede cambiar sin avisar. Si esta usando **dhcpcd** como su demonio cliente, es facil hacer un script que se ejecute cuando la dirección IP del cliente cambie. Este script se llamará por ejemplo **/etc/dhcpc/dhcpcd-eth0.exe**.

Basicamente, debe añadir una línea al script que mande una señal **SIGUSR1** o **SIGHUP** al demonio OpenVPN:

```
killall -HUP openvpn
```

Cuando OpenVPN reciba esta señal cerrará y reabrirá la conexión de red con su extremo, usando la nueva dirección IP asignada por DHCP.

También puede usar la opción **--float** si esta conectando con un extremo que puede cambiar su dirección IP debido a una reasignación DHCP.

Es también posible manejar reasignaciones DHCP con la señal **SIGUSR1** que es como **SIGHUP** excepto porque ofrece un control más detallado sobre qué subsistemas OpenVPN se recargan. Una señal SIGUSR1 puede ser también generada internamente basada en **--ping** y **--ping-restart**. La opción **--persist-tun** permite recargar sin cerrar y reabrir el interfaz TUN (que permite simular conectivad a traves del túnel con reasignaciones DHCP). La opción **--persist-remote-ip** permite preservar la dirección IP remota en reasignaciones DHCP. Esto permite a ambos extremos OpenVPN ser clientes DHCP. La opción **--persist-key** no vuelve a leer los ficheros con las claves al reiniciar (lo cual permite al demonio OpenVPN ser rearrancado incluso si sus privilegios fueron rebajados con **--user** o **--group**).

Para más información sobre como usar OpenVPN en un contexto de direcciones IP dinámicas, consulte la FAQ.

OpenVPN puede también ser usado en casos donde ambos extremos de la conexión son dinámicos.

Arrancar la VPN automáticamente al reiniciar

Primero cree un directorio para almacenar las claves y los ficheros de configuración de OpenVPN como por ejemplo **/etc/openvpn**.

Decida si desea usar el modo TLS o una clave estática y copie los ficheros apropiados **.conf**, **.up**, **.key**, **.pem** y **.crt** a **/etc/openvpn**.

Proteja sus ficheros **.key**:

```
chmod go-rwx /etc/openvpn/*.key
```

Si está usando **iptables** en Linux, edite el fichero de configuración del firewall **firewall.sh**, realizando los cambios apropiados y copielo en **/etc/openvpn**.

Haga un script de arranque similar a este:

---

sample-config-files/openvpn-startup.sh

```
#!/bin/bash

# Ejemplo de script de inicio para OpenVPN
# para Linux.

# directorio de openvpn para los ficheros de configuración
dir=/etc/openvpn

# cargar el firewall
$dir/firewall.sh

# cargar el modulo del kernel TUN/TAP
modprobe tun

# habilitar IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

# Invocar openvpn para cada túnel VPN
# en modo demonio.  También,
# puede elimiar "--daemon" de
# la línea de comandos y añadir "daemon"
# al fichero de configuración.
#
# Cada túnel debe ejecutarse en un puerto
# UDP distinto.  Use la opción "port"
# para controlar ésto.  Como todas las
# opciones de OpenVPN, puede especificar
# "--port 8000" en la línea de comandos
# o "port 8000" en el fichero de
# configuración.

openvpn --cd $dir --daemon --config vpn1.conf
openvpn --cd $dir --daemon --config vpn2.conf
openvpn --cd $dir --daemon --config vpn2.conf
```

Y haga un script de finalización similar a este:

---

sample-config-files/openvpn-shutdown.sh

```
#!/bin/bash

# parar todos los procesos openvpn

killall -TERM openvpn
```

Finalmente, añada llamadas a **openvpn-startup.sh** y **openvpn-shutdown.sh** en los scripts de arranque y parada del sistema en el directorio **/etc/init.d**.

Gestionar el arranque y la parada de multiples túneles OpenVPN

Se presenta un script de ejemplo **/etc/init.d** que creará automáticamente un túnel OpenVPN por cada fichero **.conf** que esté en **/etc/openvpn**.

Este script se instala por defecto si instala OpenVPN desde un paquete RPM.

---

sample-scripts/openvpn.init

```
#!/bin/sh
#
# openvpn       Este shell script controla el arranque y el apagado de
#               openvpn en RedHat u otros sistemas basados en chkconfig.
#
# chkconfig: 345 24 76
#
# description: OpenVPN es una aplicación para hacer túneles robusta y flexible que
#              hace uso de las características de cifrado, autenticación y certificados
#              de la biblioteca OpenSSL para asegurar redes de túneles IP utilizando
#              un puerto UDP.
#

# Contribuido al proyecto OpenVPN por
# Douglas Keller <doug@voidstar.dyndns.org>
# 2002.05.15

# Para instalarlo:
#   copie este fichero a /etc/rc.d/init.d/openvpn
#   shell> chkconfig --add openvpn
#   shell> mkdir /etc/openvpn
#   cree los ficheros .conf o .sh in /etc/openvpn (consultar más abajo)

# Para desinstalarlo:
#   ejecute: chkconfig --del openvpn

# Notas del autor:
#
# He creado un script de inicio /etc/init.d y mejorado openvpn.spec para
# registrar automáticamente el script de incicio.  Una vez que el RPM se instale
# puede arrancar o parar OpenVPN con "service openvpn start" y "service
# openvpn stop".
#
# El script de inicio hace lo siguiente:
#
# - Arranca un proceso openvpn por cada fichero .conf que encuentra en
#   /etc/openvpn.
#
# - Si /etc/openvpn/xxx.sh existe para un fichero xxx.conf entonces lo ejecuta
#   antes de arrancar openvpn (útil para hacer openvpn --mktun...).
#
# - Además para arrancar/parar puede hacer:
#
#   service openvpn reload - SIGHUP
#   service openvpn reopen - SIGUSR1
#   service openvpn status - SIGUSR2

# Modificaciones 2003.05.02
#   * Cambiado == a = por compatiblidad con sh (Bishop Clark).
#   * Si condrestart|reload|reopen|status, comprobar que ya estabamos
#     previamente ejecutando (James Yonan).
#   * Añadidas las variables lock, piddir y work (James Yonan).
#   * Si start se ejecuta dos veces, sin un stop intercalado, o
#     si se intenta start cuando el anterior start no se finalizó
#     correctamente, entonces matar cualquier proceso arrancado previamente,
#     antes de comentar la nueva operación de arranque (James Yonan).
#   * Realizar un mejor control de errores al arrancar y devolver
#     correctamente éxito o fallo al programa llamante (James Yonan).

# Ubicación del binario openvpn
openvpn=""
openvpn_locations="/usr/sbin/openvpn /usr/local/sbin/openvpn"
for location in $openvpn_locations
do
  if [ -f "$location" ]
  then
    openvpn=$location
  fi
done

# Fichero de cerrojo (lock)
lock="/var/lock/subsys/openvpn"

# Directorio de PID
piddir="/var/run/openvpn"

# Directorio de trabajo
work=/etc/openvpn

# Añadir funciones de librería.
. /etc/rc.d/init.d/functions

# Añadir configuración de red.
. /etc/sysconfig/network

# Comprobar si la red está activa.
if [ ${NETWORKING} = "no" ]
then
  echo "Networking is down"
```

```
  exit 0
fi

# Comprobar si existe el ejecutable
if ! [ -f  $openvpn ]
then
  echo "openvpn binary not found"
  exit 0
fi

# Comprobar cómo nos han llamado.
case "$1" in
  start)
        echo -n $"Starting openvpn: "

        /sbin/modprobe tun >/dev/null 2>&1

        # Desde el punto de vista de la seguridad, creo que tiene
        # sentido eliminar esto, y obligar a los usuarios que lo
        # necesiten a habilitarlo explícitamente en su scripts de arranque o
        # en la configuración del firewall.

        #echo 1 > /proc/sys/net/ipv4/ip_forward

        if [ ! -d  $piddir ]; then
            mkdir $piddir
        fi

        if [ -f $lock ]; then
            # no nos pararon correctamente
            for pidf in `/bin/ls $piddir/*.pid 2>/dev/null`; do
              if [ -s $pidf ]; then
                kill `cat $pidf` >/dev/null 2>&1
              fi
              rm -f $pidf
            done
            rm -f $lock
            sleep 2
        fi

        rm -f $piddir/*.pid
        cd $work

        # Arrancar cada .conf en $work y ejecutar .sh si existe
        errors=0
        successes=0
        for c in `/bin/ls *.conf 2>/dev/null`; do
            bn=${c%%.conf}
            if [ -f "$bn.sh" ]; then
                . $bn.sh
            fi
            rm -f $piddir/$bn.pid
            $openvpn --daemon --writepid $piddir/$bn.pid --config $c --cd $work
            if [ $? = 0 ]; then
                successes=1
            else
                errors=1
            fi
        done

        if [ $errors = 1 ]; then
            failure; echo
        else
            success; echo
        fi

        if [ $successes = 1 ]; then
            touch $lock
        fi
        ;;
  stop)
        echo -n $"Shutting down openvpn: "
        for pidf in `/bin/ls $piddir/*.pid 2>/dev/null`; do
          if [ -s $pidf ]; then
            kill `cat $pidf` >/dev/null 2>&1
          fi
          rm -f $pidf
        done
        success; echo
        rm -f $lock
        ;;
  restart)
        $0 stop
        sleep 2
        $0 start
        ;;
  reload)
        if [ -f $lock ]; then
            for pidf in `/bin/ls $piddir/*.pid 2>/dev/null`; do
                if [ -s $pidf ]; then
                    kill -HUP `cat $pidf` >/dev/null 2>&1
                fi
            done
        else
            echo "openvpn: service not started"
            exit 1
        fi
```

```
        ;;
    reopen)
        if [ -f $lock ]; then
            for pidf in `/bin/ls $piddir/*.pid 2>/dev/null`; do
                if [ -s $pidf ]; then
                    kill -USR1 `cat $pidf` >/dev/null 2>&1
                fi
            done
        else
            echo "openvpn: service not started"
            exit 1
        fi
        ;;
    condrestart)
        if [ -f $lock ]; then
            $0 stop
            # evitar condiciones de carrera
            sleep 2
            $0 start
        fi
        ;;
    status)
        if [ -f $lock ]; then
            for pidf in `/bin/ls $piddir/*.pid 2>/dev/null`; do
                if [ -s $pidf ]; then
                    kill -USR2 `cat $pidf` >/dev/null 2>&1
                fi
            done
            echo "Status written to /var/log/messages"
        else
            echo "openvpn: service not started"
            exit 1
        fi
        ;;
    *)
        echo "Usage: openvpn {start|stop|restart|condrestart|reload|reopen|status}"
        exit 1
        ;;
esac
exit 0
```

Instanciar un demonio OpenVPN usando inetd o xinetd

El servicio **xinetd** puede usarse para instanciar automáticamente un demonio OpenVPN al recibir un datagrama inicial de un extremo remoto.

La configuración de xinetd hará que xinetd escuche en el puerto UDP 1194. El primer datagrama de una sesión OpenVPN (usando una clave pre-compartida), instanciará automáticamente un demonio OpenVPN para gestionar la sesión. Nótese que el uso del parámetro de configuración **--inactive** provocará que el demonio OpenVPN dé por concluidas conexiones después de 10 minutos sin actividad. Después de que el demonio de OpenVPN termine por cualquier razón, el servicio xinetd volverá a escuchar en el puerto, y volverá a instanciar al demonio OpenVPN para manerar nuevas conexiones entrantes. También tenga en cuenta que xinetd instanciará inicialmente el demonio OpenVPN con privilegios de *root*, pero OpenVPN seguidamente (después de leer el fichero protegido con la clave) rebajará sus privilegios a *nobody*.

El fichero de claves puede ser generado con el siguiente comando:

```
openvpn --genkey --secret key
```

Tenga en cuenta que cada túnel OpenVPN necesita ejecutarse en un número de puerto distinto, y necesita su propio fichero de configuración de xinetd. Esto es porque OpenVPN necesita información específica sobre cada potencial conexión entrante, incluyendo ficheros de claves, dispositivos TUN/TAP, extremos de túnel, e información de enrutado. En este momento del desarrollo de OpenVPN, no es posible manejar ningún tipo de *plantilla de conexiones entrantes* que permita un fichero de configuración simple para describir un amplio abanico de potenciales conexiones clientes. Como OpenVPN está implementado como un servidor UDP, no puede beneficiarse de la infraestructura disponible para crear hijos en servidores TCP que escuchan en un puerto fijo, creando un nuevo demonio dinámicamente por cada sesión cliente. No obstante, las plantillas para las conexiones entrantes estan en la lista de peticiones y pueden implementarse si hay suficiente interés y soporte por parte del desarrollador y de la comunidad de usuarios.

---

sample-config-files/xinetd-server-config

```
# Un fichero de configuración de xinetd para OpenVPN.
#
# Este fichero debe ser renombrado a openvpn o algo apropiado
# y copiado al directorio /etc/xinetd.d
# xinetd puede procesar este fichero reiniciándose
# o mandandole la señal SIGHUP.
```

```
#
# Para cada cliente potencial, cree una versión separada
# de este fichero de configuración en un número de puerto único.  También tenga
# en cuenta que el tamaño de la clave y el extremo ifconfig deben ser únicos para
# cada cliente.  Esta configuración asume que el ejecutable OpenVPN
# y la clave están en /root/openvpn.  Cámbielo para ajustarlo
# a su entorno.

service openvpn_1
{
        type            = UNLISTED
        port            = 1194
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = root
        server          = /root/openvpn/openvpn
        server_args     = --inetd --dev tun --ifconfig 10.4.0.2 10.4.0.1 --secret /root/openvpn/key --inactive 600 --user nobody
}
```

---

sample-config-files/xinetd-client-config

```
# Este fichero de configuración de OpenVPN
# es la parte cliente contrario
# de xinetd-server-config

dev tun
ifconfig 10.4.0.1 10.4.0.2
remote my-server
port 1194
user nobody
secret /root/openvpn/key
inactive 600
```

Copyright (C) 2002-2004 por James Yonan <jim@yonan.net>. Este documento se distribuye bajo la GNU Free Documentation License versión 1.2. Traducido por Ramón Pons Vivanco <rpons@rinu.org>.